1. **AND**

**Input:**

import numpy as np

def activation(f):

    if f<2.5:

        return 0

    else:

        return 1

def neuron(x,w,b):

    r=np.dot(x,w)+b

    return activation(r)

def ANDgate(x):

    w=np.array([1,1])

    b=0.5

    return neuron(x,w,b)

eg1=np.array([0,0])

eg2=np.array([0,1])

eg3=np.array([1,0])

eg4=np.array([1,1])

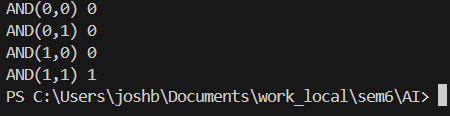
print("AND(0,0)",ANDgate(eg1))

print("AND(0,1)",ANDgate(eg2))

print("AND(1,0)",ANDgate(eg3))

print("AND(1,1)",ANDgate(eg4))

**Output:**

****

1. **OR**

**Input:**

import numpy as np

def activation(f):

    if f<1.5:

        return 0

    else:

        return 1

def neuron(x,w,b):

    r=np.dot(x,w)+b

    return activation(r)

def ORgate(x):

    w=np.array([1,1])

    b=0.5

    return neuron(x,w,b)

eg1=np.array([0,0])

eg2=np.array([0,1])

eg3=np.array([1,0])

eg4=np.array([1,1])

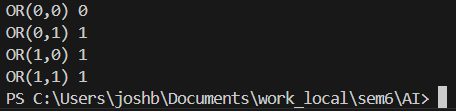
print("OR(0,0)",ORgate(eg1))

print("OR(0,1)",ORgate(eg2))

print("OR(1,0)",ORgate(eg3))

print("OR(1,1)",ORgate(eg4))

**Output:**



1. **NOR**

**Input:**

import numpy as np

def activation(f):

    if f<=0.5:

        return 1

    else:

        return 0

def neuron(x,w,b):

    r=np.dot(x,w)+b

    return activation(r)

def NORgate(x):

    w=np.array([1,1])

    b=0.5

    return neuron(x,w,b)

eg1=np.array([0,0])

eg2=np.array([0,1])

eg3=np.array([1,0])

eg4=np.array([1,1])

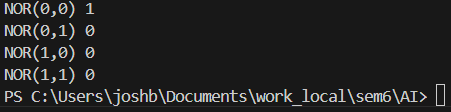
print("NOR(0,0)",NORgate(eg1))

print("NOR(0,1)",NORgate(eg2))

print("NOR(1,0)",NORgate(eg3))

print("NOR(1,1)",NORgate(eg4))

**Output:**



1. **NAND**

**Input:**

import numpy as np

def activation(f):

if f<2.5:

return 1

else:

return 0

def neuron(x,w,b):

r=np.dot(x,w)+b

return activation(r)

def NANDgate(x):

w=np.array([1,1])

b=0.5

return neuron(x,w,b)

eg1=np.array([0,0])

eg2=np.array([0,1])

eg3=np.array([1,0])

eg4=np.array([1,1])

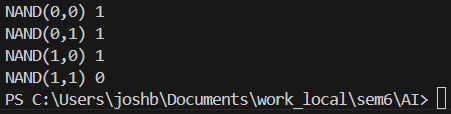
print("NAND(0,0)",NANDgate(eg1))

print("NAND(0,1)",NANDgate(eg2))

print("NAND(1,0)",NANDgate(eg3))

print("NAND(1,1)",NANDgate(eg4))

**Output:**



1. **NOT**

**Input:**

import numpy as np

def activation(f):

if f<=0.5:

return 1

else:

return 0

def neuron(x,w,b):

r=np.dot(x,w)+b

return activation(r)

def NOTgate(x):

w=np.array([1])

b=0.5

return neuron(x,w,b)

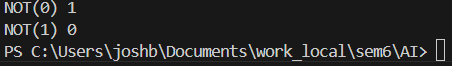
eg1=np.array([0])

eg2=np.array([1])

print("NOT(0)",NOTgate(eg1))

print("NOT(1)",NOTgate(eg2))

**Output:**



1. **3 input AND**

**Input:**

import numpy as np

def activation(f):

    if f<3.5:

        return 0

    else:

        return 1

def neuron(x,w,b):

    r=np.dot(x,w)+b

    return activation(r)

def ANDgate(x):

    w=np.array([1,1,1])

    b=0.5

    return neuron(x,w,b)

eg1=np.array([0,0,0])

eg2=np.array([0,0,1])

eg3=np.array([0,1,0])

eg4=np.array([0,1,1])

eg5=np.array([1,0,0])

eg6=np.array([1,0,1])

eg7=np.array([1,1,0])

eg8=np.array([1,1,1])

print("AND(0,0,0)",ANDgate(eg1))

print("AND(0,0,1)",ANDgate(eg2))

print("AND(0,1,0)",ANDgate(eg3))

print("AND(0,1,1)",ANDgate(eg4))

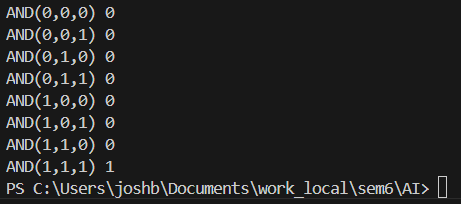
print("AND(1,0,0)",ANDgate(eg5))

print("AND(1,0,1)",ANDgate(eg6))

print("AND(1,1,0)",ANDgate(eg7))

print("AND(1,1,1)",ANDgate(eg8))

**Output:**



1. **3 input OR**

**Input:**

import numpy as np

def activation(f):

if f<1:

return 0

else:

return 1

def neuron(x,w,b):

r=np.dot(x,w)+b

return activation(r)

def ORgate(x):

w=np.array([1,1,1])

b=0.5

return neuron(x,w,b)

eg1=np.array([0,0,0])

eg2=np.array([0,0,1])

eg3=np.array([0,1,0])

eg4=np.array([0,1,1])

eg5=np.array([1,0,0])

eg6=np.array([1,0,1])

eg7=np.array([1,1,0])

eg8=np.array([1,1,1])

print("OR(0,0,0)",ORgate(eg1))

print("OR(0,0,1)",ORgate(eg2))

print("OR(0,1,0)",ORgate(eg3))

print("OR(0,1,1)",ORgate(eg4))

print("OR(1,0,0)",ORgate(eg5))

print("OR(1,0,1)",ORgate(eg6))

print("OR(1,1,0)",ORgate(eg7))

print("OR(1,1,1)",ORgate(eg8))

**Output:**

